

AFRL-IF-RS-TR-2006-92
Final Technical Report
March 2006



KNOWLEDGE CREATION TOOLS FOR DAML

SRI International

Sponsored by
Defense Advanced Research Projects Agency
DARPA Order No. K533

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the U.S. Government.

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

STINFO FINAL REPORT

This report has been reviewed by the Air Force Research Laboratory, Information Directorate, Public Affairs Office (IFOIPA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

AFRL-IF-RS-TR-2006-92 has been reviewed and is approved for publication.

APPROVED: /s/

MARK GORNIAK
Project Engineer

FOR THE DIRECTOR: /s/

JAMES W. CUSACK
Chief, Information Systems Division
Information Directorate

REPORT DOCUMENTATION PAGE			<i>Form Approved</i> <i>OMB No. 074-0188</i>	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE MARCH 2006	3. REPORT TYPE AND DATES COVERED Final Jun 2000 – Oct 2005	
4. TITLE AND SUBTITLE KNOWLEDGE CREATION TOOLS FOR DAML			5. FUNDING NUMBERS C - F30602-00-C-0168 PE - 62301E PR - DAML TA - 00 WU - 05	
6. AUTHOR(S) David L. Martin, Grit Denker, Richard Waldinger, Jerry Hobbs				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) SRI International 333 Ravenswood Avenue Menlo Park California 94025			8. PERFORMING ORGANIZATION REPORT NUMBER N/A	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Defense Advanced Research Projects Agency AFRL/IFSB 3701 North Fairfax Drive 525 Brooks Road Arlington Virginia 22203-1714 Rome New York 13441-4505			10. SPONSORING / MONITORING AGENCY REPORT NUMBER AFRL-IF-RS-TR-2006-92	
11. SUPPLEMENTARY NOTES AFRL Project Engineer: Mark Gorniak/IFSB/(315) 330-7724 Mark.Gorniak@rl.af.mil				
12a. DISTRIBUTION / AVAILABILITY STATEMENT APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED				12b. DISTRIBUTION CODE
13. ABSTRACT (Maximum 200 Words) This report summarizes the work done by SRI International for DARPA's DAML research program during the period 2000-2005. This work falls into six categories. Using a first-order logic theorem prover, we verified and refined the axiomatic semantics of DAML+OIL and OWL. We led collaborative efforts to develop foundation ontologies of time and space for the Semantic Web. We led collaborative efforts to develop ontologies and related technologies for describing and reasoning about services on the Semantic Web (this work was focused around the OWL-S and SWSF ontologies), and we investigated techniques for automated discovery and composition of Web services. We developed ontologies and related technologies for describing security policies and trust policies on the Semantic Web. We built tools for developing Semantic Web content. Finally, we developed several application systems illustrating some of these technologies.				
14. SUBJECT TERMS DAML, DAML+OIL, OWL, OWL-S, OWL-Time, Semantic Web, Semantic Web services, Semantic Web policies, Semantic Web ontologies, axiomatic semantics				15. NUMBER OF PAGES 26
				16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT UL	

Contents

1	Introduction	1
2	Axiomatic Semantics of DAML and OWL	3
3	Foundation Ontologies	3
3.1	OWL-Time: A Temporal Ontology for the Semantic Web	3
3.2	An Ontology of Spatial Relations for the Semantic Web	5
4	Semantic Web Services	6
4.1	OWL-S	7
4.2	Semantic Web Services Framework (SWSF)	8
4.3	Automated Discovery and Composition of Web Services	10
5	Security, Trust, and Policies	10
5.1	Security Ontologies	11
5.2	Security for Semantic Web Services	11
5.3	Security Services	11
5.4	Policies	12
6	DAML Tools	14
6.1	DAML+OIL Editor	14
6.2	OWL-S Editor	15
7	Applications	17
7.1	Application: Question Answering	17
7.2	Application: Travel	17
7.3	DAML-OAA Bridge Agent	17
8	Summary	18
8.1	Lessons Learned	19
8.2	Future Work	20
	References	21

1 Introduction

This report describes SRI's participation in the DARPA Agent Markup Language (DAML) research program. SRI's DAML project began in 2000 and continued through 2005.

DAML was an ambitious, visionary program to develop the foundations for the *Semantic Web*, which may be viewed as the next generation of the Internet. The Semantic Web is a set of technologies designed to enable a variety of *automated* uses of the Internet, in distinction to the current Web infrastructure, which primarily supports humans in perusing textual and graphical materials online.

The high-level objectives of DAML included:

- Develop a language and tools to facilitate the concept of the Semantic Web
- Support WWW content that is easily used by intelligent agents and other programs (annotate Web resources with machine-understandable indication of content, retrieve appropriate resources according to content, compose resources to achieve complex tasks)
- Build on World Wide Web Consortium (W3C) standards XML and RDF, as well as relevant research efforts from Knowledge Representation, Computational Logic, and other relevant fields.

In accordance with these objectives, the primary research thrusts of the program were language development, ontology creation, tool construction, technology transition, and experimentation and prototypes. This report details the contributions made by SRI in each of these areas. Here we provide a brief overview and roadmap:

- **Language development.**

At the heart of the Semantic Web is the publication of content and metadata in a well-understood knowledge representation (KR) formalism, which can support a variety of reasoning tasks. This content and metadata are captured in ontologies and knowledge bases expressed in one or more KR languages. The premiere achievement of DAML has been the definition and standardization of the Web Ontology Language (OWL). SRI contributed to this effort by developing, validating, and refining the axiomatic semantics of OWL and DAML+OIL (the pre-standardization version of OWL). This effort included simplification of axioms, elimination of redundant axioms, identification of inconsistencies and missing axioms, and validation using theorem-proving techniques. In addition, we provided a critique of DAML+OIL based on our analysis of the axiomatic semantics, and contributed important requirements to the OWL design effort, based on our experience with building ontologies in OWL. These contributions are described in Section 2 of this report.

- **Ontology creation.**

SRI took the lead in developing important ontologies for publication and use on the Semantic Web. These included foundational ontologies for representing

temporal and geospatial knowledge, as well as ontologies for describing Web services and policies related to security and trust. The work on Web services led to the preparation of OWL for Services (OWL-S) and the Semantic Web Services Framework (SWSF), both of which were submitted to the W3C. In each case we led collaborative, consensus-based efforts and drew on the most relevant prior research. These efforts are described in Sections 3, 4, and 5 of this report.

- **Tool construction.**

User-friendly, robust, effective tools are essential for the widespread adoption of Semantic Web technologies, and have been strongly emphasized during the program. In this area, SRI contributed primarily by developing a DAML+OIL editor and an OWL-S editor. Section 6 describes these tools.

- **Technology transition.**

The DAML program has been notable for its efforts to transition OWL and related technologies into the public and commercial realms. From early in the program, there was a fruitful partnership with the W3C to delineate requirements associated with bringing semantics to the Web, develop a vision and roadmap with widespread support, and polish and publish mature technologies using standardization procedures. SRI contributed to this aspect of the program in a number of ways. In connection with our work on Semantic Web services, we led the efforts to prepare the W3C submissions of OWL-S and SWSF. These submissions are discussed in Section 4. In addition, the other ontology development efforts described herein have also produced results that are publically accessible and reusable. The tools developed in this project, described in Section 6, have been made available in open-source releases on semwebcentral.org. We have also been active in presenting tutorials, organizing workshops, and publishing our results in conferences and journals, as noted in several different sections.

- **Experimentation and prototypes.** In addition to languages, ontologies, and tools, the program has also recognized the importance of developing and demonstrating valuable applications based on these technologies. SRI's work in this area has included the development and demonstration of QUARK, a question-answering system that retrieves information from the Semantic Web (as well as other kinds of sources), a travel-planning demo, and a "bridge" component that allows for DAML+OIL to be used with an agent-based system framework. These three systems are described in Section 7. In addition, Section 5 describes a system that implements matching of service security parameters, and Section 4 describes a system that implements a theorem-proving-based approach to discovery and composition of Web services.

2 Axiomatic Semantics of DAML and OWL

The DAML+OIL language was defined formally by both an axiomatic and a model-theoretic semantics. We used the SNARK theorem prover [19] to perform a validation of the axiomatic semantics ([4] and <http://vistology.com/daml/HotVIS.htm>).

Software was developed to translate the axioms for DAML+OIL from their original language, KIF, into the language of Kestrel's Specware environment. Specware was employed for its theory-management capabilities; the theory could be decomposed into components, manipulated, and verified with a single instruction. An interface developed by Kestrel translated the axioms into the language of SNARK.

Two primary methods were used in the validation. On the one hand, the theorem prover was applied to derive consequences from the axioms, by forward inference, to see if any surprising or unintended consequences were obtained. On the other hand, the theorem prover was applied to establish the validity of intended consequences, those we expected to hold.

The results of the forward inference were surprising: three inconsistencies were discovered. One of these inconsistencies actually depended on an inconsistency in the axioms for KIF, on which the DAML+OIL axioms were based. These axioms imply that the domain has only a single element; hence $0 = 1$ and $T = \text{NIL}$! The KIF axioms had been published and were available for scrutiny on the Web since 1998, but apparently had never been subjected to formal validation. Forward inference also led to the discovery of several new properties of the language. The results of the verification of intended consequences were also surprising: we discovered about 20 missing axioms.

When the new language OWL was defined, we were asked to create the axiomatic semantics for OWL-Full. We developed a set of axioms that were validated as they were being produced. They were subjected to the same tests as were the axioms for DAML+OIL. For the intended consequences, we used published test cases proposed for OWL. Although this validation process does not provide absolute certainty, we can at least be sure that the axioms are not so obviously flawed as the ones for the original language were. (See www.ai.sri.com/daml/owl/axiomatic.htm.)

SNARK was written by Mark Stickel. Specware is a product of the Kestrel Institute. The Specware-to-SNARK interface was developed, in different incarnations, by Nikolaj Bjorner and David Cyrluk. The axioms were developed and validated by Richard Waldinger. This work was supported in part by the Kestrel Institute through a DAML subcontract.

3 Foundation Ontologies

3.1 OWL-Time: A Temporal Ontology for the Semantic Web

At the spring 2002 DAML PI meeting, Jerry Hobbs led a breakout session organized around a discussion of what classes of concepts needed to be in a widely accepted temporal ontology for the Semantic Web. The following six classes were identified:

1. Topological relations, like “before” and Allen’s interval relations
2. Measures of durations
3. Clock and calendar terminology
4. Deictic terms, like “now” and “today”
5. Temporal aggregates, such as “every third Monday of every other month”
6. Vague descriptions of temporal information

It was agreed that the first three of these were the critical needs. Hobbs then wrote an axiomatization in first-order logic of these three areas. This went through several rounds of criticism and revision before it stabilized around the beginning of 2003.

One of the key innovations in the approach was to avoid “black hole” issues by keeping the ontology neutral with respect to several problematic issues, e.g., whether the endpoints of an interval are part of the interval. Another key innovation was to isolate controversial but possibly useful treatments of phenomena by axiomatizing them, but making the axioms conditional on “trigger conditions”. Thus, one could have a theory that says there are points at infinity simply by asserting the proposition “(PtsAtInfinity)”, but if one doesn’t assert that, those axioms will not come into play.

In April 2002 Hobbs began to collaborate with James Pustejovsky of Brandeis on his ARDA-sponsored effort to develop a temporal text annotation language TimeML. OWL-Time and TimeML were brought into line with each other, so that insofar as temporal information in text could be extracted as TimeML annotations, then it could be reasoned about in OWL-Time. This collaboration formed the basis of a project on “Annotating and Reasoning about Time” in Phase II of the AQUAINT program.

In the spring of 2003 Feng Pan, a student of Hobbs’s at the University of Southern California, developed a time zone resource in OWL for the entire world. The full OWL-Time was in fact expressed in first-order logic. In 2003 we produced an “entry ontology” expressed in OWL, covering the principal concepts one would need in simple applications, and OWL declarations of the concepts. Since that time we have developed treatments of the final three domains of temporal ontology – deictic terms, temporal aggregates, and vague descriptions. These have not been made part of the basic ontology yet.

In our work on temporal aggregates, we have tried to cover as wide a range of natural language examples as possible, as well as subsuming the coverage of temporal aggregates in iCalendar. We are also developing a treatment of the more problematic cases in temporal arithmetic, involving mixings of the year-month system and the week-day-hour-minute-second system.

Concerning vague descriptions, we have launched an effort to annotate events in news with the range in which their duration is likely to fall. We have developed

annotation guidelines to disambiguate the most common uncertain cases, and we have examined issues of inter-annotator agreement.

OWL-Time is now the basis of a W3C proposal, and is well on its way to becoming a recommendation. It has been used in over a dozen applications unrelated to the DAML program by various researchers around the world, indicating that it is beginning to gain acceptance. In at least one evaluation it was rated more highly than other time ontologies examined.

The papers we have published on OWL-Time and related issues are [11, 13, 17, 12].

3.2 An Ontology of Spatial Relations for the Semantic Web

At the spring 2003 DAML PI meeting, we began an effort to develop a widely accepted ontology of spatial relations for the Semantic Web, similar to our OWL-Time effort.

The aim of this ontology was to provide a way for different spatial reasoning engines and spatial resources to communicate with each other, as well as a way for people to mark up the spatial information on their web sites. The goals of the effort were to produce an ontology that would:

1. Enable general, though not necessarily efficient, reasoning about spatial concepts
2. Link with more efficient specialized reasoning engines for spatial reasoning
3. Link with the numerous databases that exist containing a wealth of specific, e.g., geographical, spatial information
4. Support convenient query capabilities for spatial information.

The topics we intended to cover included topological relations (e.g., the region connection calculus RCC8), dimension, orientation, shape, measures like length, area and volume, latitude, longitude and elevation, and political subdivisions.

Much of the work was to be focused on geographical knowledge, but the intent was not to restrict ourselves to this domain alone. Topological spatial relations are important in microbiology, for example. Other application areas that were suggested are the geology of earthquakes, NASA applications, computer graphics, and virtual reality.

In early 2004 we were able to do a certain amount of work on this ontology. We collected lists of requirements from a number of sites. We developed a first rough cut of an ontology of the topological spatial relations. Our ontology for space was introduced, not simply to represent the facts, but also to facilitate querying and the composition of information services. We were interested in deductive inference (theorem proving) over the ontology and the extraction of answers to queries from mathematical proofs. These applications were supported by two mechanisms: *answer-extraction* from proofs and *procedural attachment*. The answer-extraction mechanism allows us to find an entity satisfying desired properties by proving the

existence of an entity that satisfies those properties; the proof then determines a method for finding such an entity. The procedural-attachment mechanism lets us link a symbol in the ontology to an external procedure, so that the procedure can be executed as needed, providing more information or performing a computation, as the proof is in progress.

The ontology for space, like that for time, was developed in a sorted first-order logic, because of the lack of a suitable DAML rule language at the time the work was being conducted. A central issue was the fact that there are many names that apply to a given place, and many places that may be given the same name. For instance, there are at least six cities named Kabul in the world; the one in Afghanistan is also locally known as Kabol.

We introduced a hierarchically structured naming scheme and related this scheme to other geographical reference devices, such as latitude/longitude pairs and bounding boxes. Over a hundred geographical feature types were introduced as sorts in this ontology.

To do a thorough spatial ontology would be an immense job, but we intended to limit the effort to *linking* with resources, rather than *duplicating* them. For example, the ontology would interface with a resource on the shapes of geographical regions, but not encode its internal representations. Procedural attachments were introduced linking the ontology to the DAML representation of the CIA World Fact Book and the Alexandria Digital Library Gazetteer. Thus, the populations of countries, their borders, and their capital cities were not represented explicitly in the ontology. Rather, the information was obtained as needed by accessing the Fact Book. Similarly, the procedural attachment to the Gazetteer represented the latitudes and longitudes and other information about six million place names. Interoperability issues such as conversion between different representations were handled by procedural attachments to conversion software. For instance, unit conversion (e.g. miles versus kilometers) was handled by a procedural attachment to the multiplication function. Composition of information services was achieved by the answer-extraction mechanism.

For instance, to find the distance between Kabul, Afghanistan, and Philadelphia, Pennsylvania, the theorem prover might consult the procedural attachment to the Gazetteer to find the latitude and longitude of the two cities. It would then invoke software to compute the distance between the two lat/long pairs. Other procedural attachments would convert representations of latitudes and longitudes from that returned by the Gazetteer to that accepted by the distance software [21].

The formalization of spatial knowledge was developed by Jerry Hobbs and Richard Waldinger.

4 Semantic Web Services

DAML-funded work on Semantic Web services (SWS) has been vigorous and collaborative, and has achieved significant visibility and usage both within and outside the DAML program community. SRI has had a leadership role in this area throughout the DAML program.

Ontology development work has taken place within two collaborative forums: the OWL-S Coalition and the Semantic Web Services Language (SWSL) committee. SRI has provided leadership in both of these efforts. In addition, SRI has made significant contributions in the area of automated discovery and composition of SWS, security and privacy for SWS, and tools for Semantic Web services. SRI has also been active in pre-standardization activities, at the World Wide Web Consortium (W3C), related to Semantic Web services.

The work on ontologies, pre-standardization activities, and discovery and composition are described in this section. The work on security, privacy, and tools for Web services are described in later sections.

4.1 OWL-S

The OWL-S Coalition is a group of DAML-funded researchers who have worked together since early 2001 to produce seven releases of DAML-S/OWL-S. DAML PI Jim Hendler was instrumental in shaping the vision for this work. As of this writing, the Coalition is still working together to support the community of users, and is preparing a 1.2 version for release in January 2006.

In the OWL-S Coalition, SRI has been active by making important technical contributions, serving as chief editor of the OWL-S releases, maintaining the services section of the DAML Web site (<http://www.daml.org/services>), responding to queries and suggestions from third parties, organizing weekly teleconferences, and maintaining the agenda of the Coalition.

The work on OWL-S (and also on SWSF, described below) has been motivated by the observation that while the evolution of Web services technology is of central importance to the future of the Web, nevertheless the technologies currently under development by commercial interests do not provide a good foundation for reasoning or automation. Indeed, they contain limitations that lead to very labor-intensive approaches to the development of service-based systems. OWL-S provides the means by which to describe Web services more completely, more expressively, and with semantically well-founded underpinnings. This approach seeks to enable greater automation of many important tasks related to Web service provision and management, including development tasks, verification, discovery and selection, contracting and negotiation, mediation, composition, and monitoring. To date the greatest attention has been focused on discovery and composition.

OWL-S (formerly DAML-S) is an ontology, expressed in OWL-DL. The ontology is divided into three parts. In brief, the *profile* is used to express “what a service does”, for purposes of advertising, constructing service requests, and matchmaking. The *process model* describes “how it works”, to enable invocation, enactment, composition, monitoring and recovery. The *grounding* maps the constructs of the process model onto detailed specifications of message formats, protocols, and so forth (normally specified using the Web Services Description Language).

OWL-S represents one of the most sustained and visible ontology development efforts that has resulted from DAML, and is also the most mature body of work in SWS. During the program, this work has resulted in a great deal of valuable feedback

to the developers of OWL, and has helped to motivate the development of related technologies such as SWRL. OWL-S has been used, analyzed, and extended by a great many researchers beyond the Coalition and the DAML program. A number of tools have been built to support the use of OWL-S in various tasks (both by DAML researchers and others), including the OWL-S Editor described in a later section of this document.

The 1.1 release was packaged up and submitted to the World Wide Web Consortium (W3C) as a member submission, with a total of nine W3C members signing on as sponsors. The submission was accepted and published, as of Nov. 22, 2004, at <http://www.w3.org/Submission/2004/07/>. SRI took the lead in making this happen, and joined the W3C so as to make this possible.

During the course of the DAML program, SRI participated in a number of events related to Semantic Web services technology. Several of the most significant are mentioned here. In May 2003, we presented a tutorial on SWS at the Semantic Web for Military Users (SWMU) workshop organized by DARPA. In April 2005, we presented an updated SWS tutorial at the SWANS workshop, also organized by DARPA. (The SWANS presentation slides appear at <http://www.daml.org/meetings/2005/04/pi/SWS.pdf>). In June, 2005 Martin was a co-chair of the Workshop on Frameworks for Semantic Web services (FSWS), organized by the W3C and held in Innsbruck, Austria. The purpose of this workshop was to gather input regarding the need, requirements, and maturity of SWS technology and make recommendations to the W3C regarding possible standardization activities in this area. The co-chairs' report, authored by Martin and Steve Battle of HP, appears at <http://www.w3.org/2005/01/ws-swsf-cfp.html>. Position papers, presentation slides, and other workshop materials are also available there.

In addition to the W3C workshop, SRI took the lead in organizing several other workshops on Semantic Web services, including workshops at ISWC and the WWW conference. Research based on OWL-S was broadly represented at each of these, and also in services tracks within ISWC. OWL-S publications in which SRI had authorship include [6, 16, 5, 2, 1].

Additional information about the technical content of these efforts can be found at the OWL-S Web site (<http://www.daml.org/services/owl-s>), and in the SWS briefings that were presented at the DAML PI meetings. The Web site includes a listing of third-party research publications based on OWL-S as well as a listing of tools that are available for use with OWL-S.

Our work on OWL-S was led by David Martin, who was a founding member and acting chair of the OWL-S Coalition. Jerry Hobbs, Grit Denker, and Srini Narayanan also participated in the OWL-S Coalition for SRI. David Martin handled the pre-standardization activities.

4.2 Semantic Web Services Framework (SWSF)

The Semantic Web Services Initiative (SWSI) was envisioned, and the groundwork laid, in the latter half of 2002. DAML PI Murray Burke was instrumental in creating this vision and initiating the formation of the organization. Its primary motivation

was to promote collaboration between American and European researchers in SWS, and to lead towards a broader, more international consensus around core technology proposals. Two committees — one for architecture and one for language — began their work early in 2003. As mentioned above, SRI had a leading role in the Semantic Web Services Language (SWSL) committee. This committee included several DAML-funded participants, but also involved a number of researchers who had no DAML funding (both from America and from Europe).

SWSL chose to build out from the work on OWL-S, both in terms of greater language expressiveness and in terms of a more comprehensive ontology. A key motivation was one of the lessons learned by the OWL-S Coalition, that OWL is not sufficiently expressive to capture all aspects of process modeling. This observation also extended to aspects of contracting and negotiation. Accordingly, SWSL concentrated on the use of both first-order logic (FOL) and logic programming (LP) languages for specifying service ontologies and descriptions.

The output from SWSL is called the Semantic Web Services Framework (SWSF). SWSF defines a language (SWSL) and an ontology (SWSO) for use in describing Web services. In addition to the language and ontology documents, it also includes an overview document, a document containing illustrative application examples, and several appendices containing technical details.

The language includes a FOL component, SWSL-FOL and an LP component, SWSL-Rules, with a common syntax. The ontology is based primarily on OWL-S and on the Process Specification Language (PSL) and is expressed both in SWSL-FOL and in SWSL-Rules. The complete axiomatization, given in SWSL-FOL, is known as SWSO-FOL or FLOWS (First-Order Logic Ontology for Web Services). The SWSL-Rules version, known as SWSO-Rules or ROWS (Rules Ontology for Web Services), was derived from the complete axiomatization, and is of necessity somewhat weakened, due to expressiveness limitations inherent in LP languages. In addition, some work was done to specify methods for interoperation between a knowledge base expressed in SWSL-FOL and one expressed in SWSL-Rules.

SWSF, like OWL-S, has been packaged up and submitted to the World Wide Web Consortium (W3C) as a member submission, in May 2005, with a total of six W3C members signing on as sponsors. The submission was acknowledged and published at <http://www.w3.org/Submission/2005/07/> in September 2005. The SWSF documents are also available at <http://www.daml.org/services/swsf/>. The SWSL committee home page, which includes some related materials, is available at <http://www.daml.org/services/swsl/>. SWSF was presented and discussed at the W3C FSWS workshop mentioned above. SWSL has been identified as a source of ideas for the Rule Interchange Format Working Group at W3C, which began operating in December 2005.

Our work on SWSF was led by David Martin, a co-chair of the SWSL committee. David Martin also handled the pre-standardization activities.

4.3 Automated Discovery and Composition of Web Services

We have been investigating the automation of the process of discovery and composition of services in OWL-S and other Web service frameworks. This involves finding the services necessary to perform a given complex task and determining how to compose those services to accomplish the task. Our approach is based on the same technology we have used for question answering: extracting answers from proofs. However, because Web services can involve changes of state, we here employ a situational logic, i.e. a first-order theory in which states are represented explicitly by terms in the language.

Web service descriptions are translated into axioms in situational logic that serve to describe the capabilities of the service in the context of an axiomatic Web-service theory. The theory also describes the definitions of and relationships between the concepts used to describe services. We employ

- value axioms which specify the output or value of the service.
- effect axioms which specify the effects or changes of state the service produces.
- frame axioms which specify relations that the service does *not* change.
- ramification axioms which specify the “physics” of the world, aside from the operator descriptions.

Goals are also described in situational logic and are phrased as conjectures, or theorems to be proved, in the Web service theory. Typically a conjecture establishes the existence of the state we wish to produce and the value(s) we wish to return. As the theorem is proved, a composition of Web services is generated; when the proof is complete, the corresponding composition extracted from the proof is guaranteed to satisfy the original goal.

It remains to settle the following questions:

- How to generate Web service axioms from Web service descriptions.
- How to generate the other axioms in the Web service theory from OWL ontologies.
- Issues in ensuring the efficiency of the necessary theorem proving

We have been experimenting with examples from airline reservation services. Preliminary results have been encouraging.

The work on discovery and composition of Web services was conducted by Richard Waldinger and Daniel Elenius.

5 Security, Trust, and Policies

Information security plays an increasingly critical role in society. Given the increased importance of the World Wide Web for business, industry, finance, education, government and other sectors, security will play a vital role in the success of the Semantic Web.

5.1 Security Ontologies

We investigated existing emerging standards, such as Security Services Markup Language (S2ML) and its successor the Security Assertions Markup Language (SAML), as well as AuthXML, a specification for authentication information in XML, and XML Signature.

We developed a DAML security ontology that ties in with these industry proposals. The ontology contains concepts for

1. Credentials such as identity, certificate, login, and delegation token.
2. Security techniques such as authentication, encryption, data integrity, access control, checksums, and digital signatures.
3. Policies including delegation rules, authorization rules, and data protection rules.

We consolidated our DAML security ontology in collaboration with members from the DAML community, and established the Web page <http://www.daml.org/services/owl-s/security.html> as part of the public OWL-S Web site. This security Web page gives an overview of the main security-related ontologies that have been developed as part of the DAML project. For all ontologies a brief summary is given and all ontologies have been validated using BBN's vOWLidator and Mindswap's Pellet.

5.2 Security for Semantic Web Services

In April 2003 we started expanding our security approach to Semantic Web services. The proposed framework covers annotation and matchmaking. The security ontologies developed earlier were used to describe the security requirements and capabilities of web services providers and requesting agents. A reasoning engine decides whether agents and web service have comparable security characteristics by verifying that the agent's requirements are satisfied by the web service's capabilities and the service's requirements are met by the agent's capabilities. Our prototypical implementation uses the Java Theorem Prover from Stanford, for deciding the degree to which the requirements and capabilities match based on our matching algorithm. The security reasoner is integrated with the Semantic Matchmaker from CMU giving it the ability to provide security brokering between agents and services [7].

5.3 Security Services

Another focus of our security work was design and deployment of OWL-S security services [9]. We designed several security services such as encryption and decryption service for files and text messages, signing and verification of XML documents, and generation of X509 certificates. These services have been implemented in Java using a variety of publicly available cryptographic libraries (such as BouncyCastle for RSA encryption). An encryption/decryption service has been deployed as a web service.

For this purpose we employed the publicly available Web Service Toolkit (WSTK) from IBM (<http://alphaworks.ibm.com/tech/webservicestoolkit>). WSTK provides the WebSphere Application Server that can host web services. WSTK also includes a UDDI server for service registry and lookup. Our security services run on the WSTK server. A client server (implemented using Apache Tomcat) offers a GUI in which the client can select the security service. In the case of the encryption/decryption service, the client can select a file or type a message and choose between encryption and decryption. The resulting encrypted/decrypted message is displayed (or, in case of file encryption/decryption, a link to the encrypted/decrypted file is provided). We provided OWL-S annotations for our service, using OWL-S profile, OWL-S process, OWL-S grounding and WSDL annotations.

5.4 Policies

We also addressed policy issues for Semantic Web services [15, 14]. Based on our work presented on security mark-up for Semantic Web services, our goal was to propose more fine-grain policies as part of the representation of Web services and, in particular, of Semantic Web services. Policies provide the specification of who can use a service under which conditions, how information should be provided to the service, and how provided information will be used later. We addressed two kinds of policies: privacy policies and authorization policies. Privacy policies specify under what conditions information can be exchanged and what are the legitimate uses of that information. Authorization policies constrain the provider to only accept requests for service from certain clients.

We also proposed an ontology and markup to capture security information of Web service input and output parameters. The problem of representing data confidentiality in the markup of Semantic Web services such as OWL-S is that encrypted data by its very nature does not reveal its internal value or structure because it is just a byte string. We therefore suggested a semantic markup that specifies the security characteristics of input and output parameters of Web services while keeping information about the structure of the data without revealing its value. This meta-information about the kind of data exchanged with a service can be used for service selection.

One particular application domain under the topic of policies for services arose from our work in the context of a project called *Semantic Firewall*, a collaboration with the Institute for Human-Machine Cognition, the University of Southampton, UK, and IT Innovations, UK.

The EU-funded Semantic Firewall project (see <http://www.csl.sri.com/users/denker/sfw/>) will develop a formal basis, methods, and tools for Grid security. The problem is to enforce network security policies between trust domains in the presence of dynamically changing and unpredictable Grid communication needs.

In order to specify Grid service workflows, one needs to be able to describe the interaction protocol of a grid service with clients and other services. [3] defines a model to describe the interactions of multiple parties with respect to a single Grid service, distinguishes between interactions initiated by the Grid service itself and

interactions that are initiated by clients or other cooperating services, and captures state changes within the service. This model allows for the representation of inherent security policies of Grid services (see also www.csl.sri.com/users/denker/sfw/). The protocol defines what are the appropriate messages that can be exchanged at any given moment from any of the parties involved in the interaction, and it is service-centric in the sense that we focus on just those interactions that include the service we wish to describe.

In collaboration with Santtu Toivonen from VTT, Finland, we worked on trust issues for communication [20]. Messages conveyed to us in everyday life often have an influence on our decisions and behaviors. The more trustworthy a message is considered, the higher will be its impact on the recipient. Trust is a very significant commodity in communication.

We argue that for message-based communication, a more general notion of context information needs to be taken into consideration to determine the trustworthiness of a message. The decision whether or not to trust a piece of data can depend on many factors including the creator (who) of the data (what), time (when), location (where), the intent (why) of origination, the social context of the receiver, and many others. Generally, context information can have an impact on trust-relevant aspects of communications. Another advantage of making context explicit in message exchanges is that this information can be used to derive trustworthiness assertions (see <http://www.daml.org/services/owl-s/security/context/>).

Under the general topic of policies, we also studied privacy policies for OWL-S Web services. Inspired by approaches such as P3P, Rei, and KAoS, our goal was to provide means for protecting personal and enterprise privacy in the interaction of SWS and intelligent agents. Our approach allows each intelligent agent as well as SWS to have their privacy policies attached to them. We make the exchange of privacy policies between agent and service a part of the process of setting up the connection. The agent is required to meet the privacy set forth by the service's authority and vice versa. As we allow for services to communicate with other services, we use the notion of "provider" and "requester" to refer to the communicating parties.

We defined a privacy ontology that supports specification of policies such as the those illustrated above. The ontology and example policies can be found at <http://www.daml.org/services/owl-s/security.html>. In addition to the policies, we designed an algorithm for privacy policy matching. The algorithm is used in a protocol between provider and requester to assure that both parties fulfill the requirements of the other party. If the policies of the client and the service do not match, that is, one or more requirements of either party are not fulfilled by the other party, no communication will be established. If the policies match, then in principle a communication between the two communicating parties can be established. There is still more to do to enforce the policies. For example, required encryption mechanisms have to be put into place. For this one could imagine employing the security services that we implemented and described earlier.

We also wrote a position paper [8] discussing language requirements for privacy policies and how rules can be used to define the meaning of privacy policies. Once we understand the issues involved in expressing the meaning of privacy policies, we can think about translations into existing frameworks or languages to make use of their reasoning and other tool capabilities.

We were interested in expressing policies using semantic markup languages such as OWL and SWRL and in investigating whether such policies can be analyzed using existing specification and analysis frameworks. In particular, we investigated the use of the rule-based framework Maude for this purpose [18]. Many SWS policies can be formalized as a collection of rules and often policies abstractly define the model of the service in terms of a state transition system where the rules govern how a state changes. Therefore, we proposed to use a rule-based analysis framework such as Maude in conjunction with the expressiveness of semantic markup languages such as OWL and SWRL to yield a formal framework which can consume machine readable policies and support analysis of policies with respect to consistency and reachability of certain states and other analysis checks defined below. The advantage of existing rule-based formal frameworks is that they usually already support a variety of analysis tasks that can be adapted to the task at hand.

Grit Denker led SRI's DAML work on security ontologies, security for Semantic Web services, security services, and policies. In collaboration with Jon Pearce from San Jose State University, the SRI student interns Andrew Ton, Son Nguyen, and Rukman Senanayake helped with the design and implementation of security services, privacy policies, and rule-based approaches to policies. Substantial collaborations on policies included the groups of Katia Sycara at CMU, Tim Finin at UMBC, Jeff Bradshaw at IHMC, Terry Payne at University of Southampton, UK, and Mike Surridge, IT Innovations, UK. Moreover, we collaborated with Santtu Toivonen from VTT, Finland on trust.

6 DAML Tools

6.1 DAML+OIL Editor

Our goal was to provide user-friendly software that enables creating and manipulating DAML ontologies and their instantiations. We investigated the existing tools for ontology editing and processing and decided to implement a DAML+OIL plug-in for Protege. This way we could take advantage of the sophisticated graphical interface of Protege, the open-source code and the modular Protege architecture that encourages plug-ins. We implemented a DAML+OIL backend that enables the specification of ontologies and instances in the Protege editor and exports them to DAML+OIL (version published in March 2001).

By linking DAML and Protege, we made DAML+OIL accessible to the users of Protege, who number in the hundreds or perhaps thousands, so that this work, in addition to being a necessary step in the development of the resource tool, was also an important means of promoting DAML.

There was considerable interest shown by members of the Semantic Web community to use our tool. We released a version of the DAML+OIL plugin for Protege-2000. The plug-in allows users to read, edit, and generate DAML+OIL ontologies. Detailed information about the plugin, how to use it, and its current functionality can be found at <http://www.ai.sri.com/daml/DAML+OIL-plugin/>.

We continuously supported our growing user base with technical help and user guidance. Though we do not have an exact user head count, the mailing list indicates that over 100 users are interested in the plugin. Moreover, we got an average of 1-2 new users a week inquiring about the plugin.

6.2 OWL-S Editor

In late 2003 we started developing concepts for an OWL-S Editor and in 2004 and 2005 we designed and implemented an OWL-S editor as a plugin to Protege [10]. The software is available as open-source at <http://owlseditor.projects.semwebcentral.org/index.html>. Documentation is also available at that URL.

We designed a Protege GUI for specifying service interfaces to support all four modes (grounding, process, profile, service) in an OWL-S tab. The OWL-S Editor supports all OWL-S language concepts and has many features, including:

- **Highlighting related specification elements.** That is, for a selected mode (e.g., service), all other instances that are connected via properties (e.g., the profile that presents the service or the process that describes the service) are highlighted.
- **Specialized, service-specific slots widgets.** These widgets support user-friendly list representation, parameter specification (such as wrapping class names in URL), as well as editing of expressions.
- **Process editor - control flow.** We proposed and implemented a design that allows the user to view and edit the process control flow as a tree. In particular, we provided a visual drag and drop composite process editor. Additionally, we support the visualization of the control flow as a graph in the OWL-S Editor.
- **IOPR Manager.** We designed an Input-Output-Precondition-Result manager that maintains IOPR relationships between OWL-S subontologies and performs consistency checks. The consistency checks are done between the IOPRs of a process and its corresponding process. If one defines IOPRs in the profile that are not in the corresponding process, the consistency check will return with appropriate error messages and suggest, if possible, fixes. In addition to checking consistency of IOPR definitions between profiles and processes that are related via the `process:has_process` property, the consistency checker also performs other checks. For example, if one uses a class wrapped in a URI in a data type and then delete the class, the consistency checker will notice this as an inconsistency.

- **Graph Overview.** This capability visualizes and navigates the relationships between instances of the OWL-S subontologies and shows how Service/Process/Profile/Grounding instances relate to one another. The overview shows all services, profiles, top-level processes, groundings and atomic process groundings. This gives a good overview in which way the various elements are connected.
- **Generation and import of skeletal OWL-S from WSDL.** On the basis of Mindswap’s open-source WSDL2OWL-S tool, we implemented support for the creation of OWL-S from WSDL as part of our OWL-S Editor.
- **Data Flow editor.** Support for adding data flow declarations and visual representation of dataflow in the process graph was provided.
- **WSDL Widgets.** In the Grounding editor we support all three kinds of WSDL message maps through a special widget, the so-called WSDL widget. The top part of the WSDL widget can be used to map OWL-S parameters, XSLT strings or XSLT URIs to WSDL message parts. The lower part is used to map WSDL message parts, XSLT strings or XSLT URIs to OWL-S parameters. If the user chooses XSLT String for the mapping, then he will get a box in which he can type the XSLT transformation string. Similarly, the XSLT URI provides a text box in which the user can type the URI of the transformation string.
- **Service Execution** is supported from within the editor.
- **Service Matching.** A matchmaker for process composition and the ability to “wrap as composite process” by right-clicking on a node in a composite process tree, as well as “unwrapping” composite processes were integrated into the editor.
- **Integration with the SWRL editor** was provided.
- **Multiple ontology editing.** Integration of OWL-S tab with new Protege capabilities for multiple ontology editing was provided.

All OWL-S Editor features, user tutorial, installation manuals and other documentation is available online.

Grit Denker and David Martin co-led the tools initiative at SRI. SRI staff John Khouri, Shahin Saadati and Fred Gilham, and student interns Daniel Elenius, Rukman Senanayake, and John Pacheco also contributed to the design and implementation of the released systems. We collaborated with Stanford’s SMI group on the design of the OWL-S Editor.

7 Applications

7.1 Application: Question Answering

Spatial and temporal ontologies were used in the GeoLogica and QUARK systems, which were applied to question-answering for Earth systems scientists and intelligence analysts, respectively. The two systems had the same structure but invoked different ontologies. Questions were expressed in English and translated into a logical form. The query was regarded as a conjecture and submitted to the theorem prover SNARK. Snark attempted to prove the conjecture and extracted an answer to the query from the proof. Procedural attachment was used to link the base ontology to external sources, which could be in the DAML or OWL languages or in other forms. For access to DAML and OWL sources, the Agent Semantic Communications Service (ASCS) of Teknowledge was used. In particular, the DAMLized form of the CIA World Fact Book was accessed through ASCS (<http://projects.teknowledge.com/DAML/>). Temporal reasoning was accomplished through a temporal ontology that was built into SNARK. Access to intelligence reports was provided by a procedural attachment to an information extraction system.

For instance, GeoLogica could deal with questions such as “Show a petrified forest in Zimbabwe that is within 200 miles of the capital of Zambia.” Quark handled questions such as “Could Mohammed Atta have met with an Iraqi secret agent in 2001?” Access to Web-based satellite imagery and three-dimensional terrain visualizations were provided.

This work was directed by Richard Waldinger. Procedural attachments to the Alexandria Digital Library Gazetteer, ASCS, the CIA World Fact Book, and other sources, were developed by Martin Reddy and Peter Jarvis. The work was partially supported by a Cooperative Agreement from NASA, under the Intelligent Systems Program, and by a contract from ARDA, under the Aquaint program.

7.2 Application: Travel

As an application of temporal reasoning and the invocation of Web services, we demonstrated a system that could make arrangements to attend conferences. The system obtained the dates of the conference from Web sources and used Travelocity to make reservations on the appropriate dates. User preferences were modeled as to choice of airlines and hotels, number of stops, and other options.

This work was performed by David Martin and Richard Waldinger.

7.3 DAML-OAA Bridge Agent

In 2001, we completed work on a DAML-OAA (Open Agent Architecture) bridge agent, which provides an entirely new means of accessing DAML knowledge bases. The Open Agent Architecture is a framework for constructing agent-based systems, which has been under development and in use at SRI since 1994 (<http://www.ai.sri.com/oaa>). A wide variety of agent-based systems have been based upon OAA,

and it was used extensively in the successful NEO TIE (Non-combatant Evacuation Operation Technology Integration Experiment) demo constructed for the coABS program.

This bridge agent accomplishes the following goals: it allows access via OAA to knowledge (ontologies and instances) expressed in DAML; it enables integrated use of capabilities provided by OAA agents and DAML-enabled sites; and it provides a Prolog-like query language for DAML. (Requests to OAA agent communities are expressed in an Interagent Communication Language (ICL), which is based on Prolog.) At invocation, the agent is given a URL for a DAML web page. It then loads the DAML declarations into an internal triples model; this functionality is provided by HP's Jena parser. It then connects to an OAA facilitator, and registers OAA "solvable" (capabilities declarations) representing 4 different categories of DAML entities, namely DAML+OIL "meta-classes" and "meta-properties", user-defined classes, user-defined properties, and a special "low-level" solvable, `daml:triple`, that can be used to access arbitrary contents of the triples model. Each of these predicates can then be queried. For example, if the DAML-OAA agent has loaded `daml+oil-ex.daml`, the query `Class(X)` will return values for X such as `Person`, `Height`, and `BigFoot`, and `Property(X)` will return values such as `hasHeight`, `hasParent`, and `shirtsize`. `intersectionOf(TallMan, X)` will return `X = [TallThing, Man]`, the query `Person(X)` will return values such as `Peter` and `Santa`, `hasParent(Peter, Y)` will return `Y = 15`, and `daml:triple(Peter, type, Z)` will return `Z = Person`.

In addition, OAA makes it possible for multiple DAML OAA agents to be used together, through the services of a single facilitator. Following the completion of some additional work, we expect compound queries to also be supported.

This agent creates an opportunity for using DAML knowledge bases in conjunction with existing agent-based systems, and for accessing DAML knowledge bases using a Prolog-based query language.

The work on the DAML-OAA Bridge Agent was supervised by David Martin and supported by Grit Denker and student intern Imen Atallah.

8 Summary

The DAML program was motivated by an extremely compelling vision of a next-generation Internet where knowledge is represented not just as text and graphics for human consumption, but also in more formal representational schemes that enable automation and reasoning by software agents. The program has taken some solid steps towards realizing this vision, and has had a far broader impact on society at large than most programs of this sort. A full accounting of the many ways in which this impact has occurred is beyond the scope of this report. We can only mention some of the more important and visible steps that have occurred: OWL has been standardized at the W3C, several related languages have been proposed, and a working group has recently formed to standardize a "rules interchange format" building on some of this work. A number of projects and communities of interest have developed (or are developing) ontologies in OWL and made them available

for shared use. A wide variety of open-source tools have been made available for creating and publishing ontologies and knowledge bases on the Semantic Web. A number of successful technology transition demonstrations and events have occurred. Several important Semantic Web standardization activities are ongoing. A thriving worldwide research community is continuing to investigate Semantic Web research challenges. A variety of experimental systems have been demonstrated. Perhaps the most striking development of all is the number and variety of Semantic Web technology uses that were not mandated or stimulated in any way by DARPA or any other government organization.

As explained in the foregoing sections, SRI has made substantial contributions in most of these areas.

8.1 Lessons Learned

Some general observations and lessons learned from the program have been:

- Consensus and community-wide adoption of new technologies are in general difficult to achieve. Standardization and related activities at organizations such as W3C can be extremely valuable steps towards widespread adoption. It is essential to provide clear-cut demonstrations of value added to communities of interest that are likely to benefit and are positioned to adopt the new technologies. These demonstrations must be related to “real-world” use cases. It is equally critical to provide robust, easy-to-use tools. Tools and applications must be designed so as to allow for an incremental path to adoption.
- At least in the early years of the program, the language layers and versions were critical prerequisites for most other work, and were also “moving targets”. This probably resulted in some lost time and lost opportunities. Earlier development of more expressive language layers — the rules layer in particular — would likely have enabled more rapid progress in a number of areas.
- On the other hand, it must be recognized that language, tool, and application components usually develop in an iterative or spiral fashion. For example, in DAML there were a number of cases in which ontology and tool development activities provided important insights regarding language issues and requirements towards the further evolution of the language layers.
- “Complete, don’t compete.” To help ensure technology adoption, it is important to build on successful existing technologies, rather than trying to displace them. However, it is equally important to make informed choices about the suitability of the existing technologies as building blocks.
- “Don’t let the perfect be the enemy of the good” (as stated on various occasions by the DAML program managers). The success of a technology depends on many factors, not exclusively on its elegance and comprehensiveness. A number of successful technologies have the characteristic that (roughly) 20their features (or their learning curves) will cover 80needs.

8.2 Future Work

Although the DAML program has laid the foundation for a revolutionary new style of Internet use, there are a number of areas in which additional work can help ensure the success and effectiveness of these technologies going forward. These include:

- further development and standardization of more expressive language layers
- development of more effective ways of “bridging” between language layers where there are fundamental semantic mismatches (such as monotonicity vs. nonmonotonicity)
- additional work on describing more complex aspects of Web services, and evolution of the reasoning approaches needed to achieve automated contracting, negotiation, composition, monitoring, etc. of services
- more comprehensive approaches for ensuring security and privacy on the Semantic Web, and an incremental strategy towards their deployment
- more powerful technologies that mediate (translate) between heterogeneous ontologies covering the same conceptual space
- more fully developed conventions regarding architectural elements of the Semantic Web
- nurturing of collaborative interest groups to develop shared ontologies and best practices within selected domains, such as the Life Sciences group currently underway at W3C
- more compelling demonstrations of value added that speak more directly to the needs of important communities of interest
- development of better tools, or evolution of existing tools, for producing semantic content (e.g. by semi-automatic extraction from text)
- standardization, as soon as feasible, in each of the above areas.

References

- [1] A. Ankolenkar, M. Burstein, J. Hobbs, O. Lassila, D. Martin, S. McIlraith, S. Narayanan, M. Paolucci, T. Payne, K. Sycara, and H. Zeng. DAML-S: A semantic markup language for Web services. In *Proceedings of The International Semantic Web Working Symposium (SWWS)*, Stanford, CA, USA, July 2001.
- [2] Anupriya Ankolenkar, Mark Burstein, Jerry R. Hobbs, Ora Lassila, David L. Martin, Drew McDermott, Sheila A. McIlraith, Srinu Narayanan, Massimo Paolucci, Terry R. Payne, and Katia Sycara. DAML-S: Web service description for the Semantic Web. In *Proceedings of The First International Semantic Web Conference (ISWC)*, Sardinia, Italy, June 2002.
- [3] R. Ashri, G. Denker, D. Marvin, M. Surridge, and T. Payne. Semantic Web Service Interaction Protocols: An ontological approach. In S. McIlraith, D. Plexousakis, and F. van Harmelen, editors, *Proc. 3rd Intern. Semantic Web Conference (ISWC 2004)*, Hiroshima, Japan, November 2004, pages 304–319. Springer, 2004. LNCS 3298.
- [4] Kenneth Baclawski, Mieczyslaw M. Kokar, Richard J. Waldinger, and Paul A. Kogut. Consistency checking of Semantic Web ontologies. In *International Semantic Web Conference*, pages 454–459, 2002.
- [5] J. Bryson, D. Martin, S. McIlraith, and L. Stein. Toward behavioral intelligence in the Semantic Web. *IEEE Computer*, pages 48–54, November 2002.
- [6] G. Denker, J.R. Hobbs, D. Martin, S. Narayanan, and R. Waldinger. Accessing information and services on the DAML-enabled Web. In S. Decker, D. Fensel, A. Seth, and S. Staab, editors, *2nd. Intern. Workshop on the Semantic Web SemWeb'2001, Workshop at WWW10, Hong Kong, China*, May 2001. <http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-40/>.
- [7] G. Denker, L. Kagal, T. Finin, M. Paolucci, and K. Sycara. Security for DAML Web services: Annotation and matchmaking. In D. Fensel, K. Sycara, and J. Mylopoulos, editors, *Proc. 2nd Intern. Semantic Web Conference (ISWC 2003)*, Sanibel Island, FL, October 2003. Springer, 2003. LNCS 2870.
- [8] G. Denker and D. Martin. Using rules to define the semantics of privacy policies. www.w3.org/2004/12/rules-ws/accepted.
- [9] G. Denker, S. Nguyen, and A. Ton. Semantic security Web services in OWL-S. In C. Bussler, J. Davies, D. Fensel, and R. Studer, editors, *The Semantic Web: Research and Applications. 1st. European Semantic Web Symposium (ESWS2004)*, Heraklion, Crete, Greece, May 10-12, 2004, pages 240–253, 2004.
- [10] D. Elenius, G. Denker, D. Martin, F. Gilham, J. Khouri, S. Sadaati, and R. Senanyake. The OWL-S editor - a development tool for Semantic Web services. In Asunción Gómez-Pérez and Jérôme Euzenat, editors, *ESWC*, volume 3532 of *Lecture Notes in Computer Science*, pages 78–92. Springer, 2005.

- [11] J. Hobbs. Towards an ontology for time for the Semantic Web. In *Proc. LREC 2002 Workshop on Annotation Standards for Temporal Information in Natural Language, Las Palmas, Spain, May 2002*, pages 28–35, 2002. <http://www.isi.edu/~hobbs/lrec-time-may02.pdf>.
- [12] J. Hobbs and F. Pan. An ontology of time for the Semantic Web. *ACM Transactions on Asian Language Information Processing*, 3(1), March 2004.
- [13] J. Hobbs and J. Pustejovsky. Annotation and reasoning about time and events, 2003. <http://www.cs.rochester.edu/~ferguson/daml/hobbs-pustejovsky.pdf>.
- [14] L. Kagal, T. Finin, M. Paolucci, N. Srinivasan, K. Sycara, and G. Denker. Authorization and privacy for Semantic Web services. *IEEE Intelligent Systems*, 19(4):50–56, July/August 2004.
- [15] L. Kagal, M. Paolucci, N. Srinivasan, G. Denker, T. Finin, and K. Sycara. Authorization and privacy for Semantic Web services. In *Proc. of the AAAI 2004 Spring Symposium on Semantic Web Services*, March 2004.
- [16] D. Martin, M. Paolucci, S. McIlraith, et al. Bringing semantics to Web services: The OWL-S approach. In J. Cardoso and A. Sheth, editors, *Semantic Web Services and Web Process Composition: First International Workshop*, volume 3387 of *Lecture Notes in Computer Science*, page 26, San Diego, CA, USA, January 2005. Springer-Verlag.
- [17] Feng Pan and Jerry R. Hobbs. Temporal aggregates in OWL-Time. In *Proceedings, Workshop on Natural Language-based Knowledge Representations: New Perspectives*, Clearwater Beach, FL, USA, May 2005. At the Florida Artificial Intelligence Research Society International Conference (FLAIRS 2005).
- [18] R. Senanyake, G. Denker, and D. Martin. Towards integrated specification and analysis of machine-readable policies using Maude. In L. Kagal, J. Hendler, and T. Finin, editors, *Proceedings of the Semantic Web and Policy Workshop, Galway, Ireland, November 7, 2005*, 2005. www.cs.umbc.edu/swpw/.
- [19] Mark E. Stickel, Richard J. Waldinger, and Vinay K. Chaudhri. A guide to SNARK. Technical Report Unassigned, AI Center, SRI International, 333 Ravenswood Ave., Menlo Park, CA 94025, May 2000.
- [20] S. Toivonen and G. Denker. The impact of context on the trustworthiness of communication: An ontological approach. In Jennifer Golbeck, Piero A. Bonatti, Wolfgang Nejdl, Daniel Olmedilla, and Marianne Winslett, editors, *ISWC Workshop on Trust, Security, and Reputation on the Semantic Web*, volume 127 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2004.
- [21] R. Waldinger, D. E. Appelt, J. Fry, D. J. Israel, P. Jarvis, D. Martin, S. Riehemann, M. E. Stickel, M. Tyson, J. Hobbs, and J. L. Dungan. *Deductive Question Answering from Multiple Resources*. AAAI, Menlo Park, CA, 2004.